The Man Behind The Machine?

A PC Exclusive Interview With Software Guru Bill Gates

HOW WAS IBM ABLE TO SO GAUGE THE PERSONAL COMputer market as to come out with a machine that both incorporates all the good features of existing personal computers and accurately points the direction of future ones?

PC Publisher David Bunnell had a hunch that the answer to this question was to be found in Seattle—home of Microsoft, the first personal computer software company.

His hunch was based on the fact that while several software companies were chosen by IBM to provide the initial software for the IBM Personal Computer, only Microsoft provided a complete range of software. This software includes the IBM Personal Computer Disk Operating System, MACRO-assembler high-level languages BASIC, Fortran and Pascal and even application programs (Adventure and Typing Tutor).

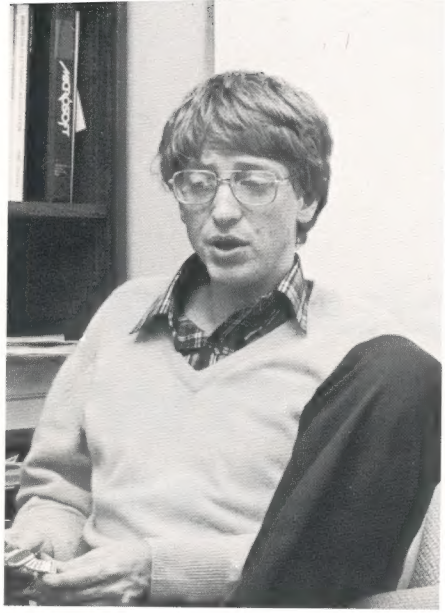
So Bunnell hopped a plane to Seattle to investigate for himself.

Sure enough Microsoft's involvement was total, day-in, day-out. For more than a year, 35 of Microsoft's staff of 100 worked fulltime (and plenty of overtime) on the IBM project. Bulky packages containing computer gear and other goodies were air-expressed almost daily between the Boca Raton laboratory and Seattle. An electronic message system was established and there was almost always someone flying the arduous 4,000 mile commute. While many other individuals and companies consulted with IBM during the course of "IBM PC" development, and most have intriguing yarns to tell, only one company worked with IBM in such an intimate and (especially for IBM) *unheard of* fashion.

by
David Bunnell

notography by Jacqueline Poitier. Color Tinting by Bennet Hall.

eprinted with permission from the February/March 1982 Issue of PC: The Independent Guide to IBM Personal Computers.


"Before they came, they said, 'Hey, we may really do some business. It could be exciting.'"

The highlight of Bunnell's investigation was a fascinating two-hour exclusive interview with Bill Gates, president and cofounder of Microsoft. As it turns out Gates probably knows more about the IBM Personal Computer and its history than anyone (outside of IBM, of course).

DAVID: What can you tell us about Microsoft's involvement with IBM on the Personal Computer project? How it was initiated and what transpired, as much as you can reveal.

BILL: In the case of the IBM project we started off not really knowing what they wanted. They came out in July of 1980 and first talked with us on a very tentative basis as though they were just doing market research. They said, "Don't get too excited and don't think anything big is going to happen."

Then they talked about how something could be done fairly quickly if a machine was designed to run standard software. In fact, we found out later that behind the scenes different labs within IBM had been charged with looking into how they could get a project done on a very quick basis. The typical product design time for a large company like IBM, and they keep track of this, is a little over four years. That is partly because they do such a complete job, and yet, in the personal computer industry, which they had a desire to participate in, you really couldn't be competitive if you speced out your product in 1976 and sold it in 1980. You would be selling an Altair computer against an Apple II.

So they wanted to come up with some way of doing things a little differently. One of the development managers of IBM got a committee together, people from different laboratories, and told them to go out and research the issue. The people we met with were from the Boca Raton laboratory, simply putting together some thoughts, essentially about how to cheat, and their idea was to use software that already existed out in the world, and to use industry standard parts like the Intel microprocessor. So they went back and said that based on using that approach they could get something done in the order of a year.

My understanding is that some of the other groups put in proposals that involved emulating existing IBM instruction sets, and there have been a lot of rumors that one of the groups looked at buying a machine from Japan. In fact, one of our Japanese customers had us do some demonstration software that was probably for that lab that was looking at Japanese sourcing.

In any case, Boca Raton got the go-ahead sometime in late 1980 and they came out with a lot of people, about 12 people. Before they came, they said, "Hey, we may really do some business. It could be exciting." And then they said, "We have a lot of things to do, we'll have our technical team meet with your technical team, so let's do them in parallel. We'll have our legal team meet with your legal team, we'll have the purchasing team meet with your purchasing team, we'll have our technical team meet with your technical team, so we can do four or five things at once." Well, that is fine, but that's me who is going to do those things and I can do only maybe two things at once, so we're not going to be able to have five simultaneous meetings.

Anyway, they came out with 12 people, and we really got things going. We ended up making the hardware a little more state-of-the-art by putting new things in it that went beyond the cardinal rule of getting the project done in a year. But, you know, the second priority beyond getting it done in a year was to have a state-of-the-art machine and by using the 16-bit processor and doing some of the stuff in the graphics, I think everyone pretty much agrees that that was achieved.

DAVID: Why is it important to have a 16-bit processor?

BILL: That is one area where there is a lot of confusion because the standard thing in the industry nowadays is to say, "Who cares about what's inside the machine?" People are buying a solution, not a computer, which is absolutely true. They are buying things like word processing or Visicalc which is one of the applications IBM announced.

I think 16-bits is extremely important, and it is not because of speed, although if you sit down at an IBM machine and play with it a little while you will see that it performs significantly better than existing 8-bit machines.

The main reason for the 16-bit micro being advantageous is its increased address space. That sounds like a technical issue, but what it boils down to for the end-user is that we can do more complex software, with a better end-user interface, in a more transportable form than we have ever been able to do in what I call the "8-bit world."

When I say 8-bit world, I mean the 6502 microprocessor, which is the chip used in the Apple, the Pet and the Atari, or the most popular chip which is the 8080, Z-80 family used in the Xerox 820 machine, the NorthStar, Vector Graphic and many others. In those 8-bit machines there is one common characteristic, which is that the logical address space in the machine is limited to 64K bytes (about 64,000 characters of storage). You have to put the operating system, the program, the data, the graphics memory, if it is going to be efficient—all those things in a single 64K area. You get into some terrible problems where you have to write program code in a hard to maintain fashion to keep it small and in fact that is one of the things that Microsoft is doing absolutely the best job of, is writing stuff in a small amount of space. Its a fine art that we spend a lot of time on, because in 8-bit machines it really made a lot of difference. But this is no longer our focus on 16-bit machines.

People also compromised in the end user interface with their packages because they simply could not get enough stuff in there, and finally the overall capability of the packages are also compromised because you want it to be able to run on all the 8-bit machines. For example, whenever we have put a new feature in BASIC such as good screen handling, which is something that we are working on now, people complain because any feature we put in takes away from the space available for an application.

Now in the 8088 (the Intel 16-bit microcomputer used by IBM), that limit, the logical address space limit, is for all practical purposes gone away. The chip is designed to address up to a megabyte (1 million characters). IBM's announced support for up to a quarter megabyte, that is 256K, and it is very much in the relevant range. In other words, that factor will make all the difference in terms of quality end user interface integrated software.

DAVID: Will your recently announced planning package, Multiplan, be integrated with word processing?

BILL: Not in its initial release. When we first get an extra resource, we don't know all the ways we are going to be able to take advantage of it. All I can really say is the 64K barrier has been the critical constraint in terms of writing software in a transportable form and putting new features in. Now that we have the freedom, we can use some more creativity to take advantage of it. It's just like high resolution graphics was on the Apple. When the Apple II first came out it had high resolution graphics, but for about three years, nobody wrote programs that would take advantage of it. The programs were low resolution and it was kind of bizzare to try to use that extra mode. But today, the Apple II is virtually defined by high resolution graphics. There is simply not an entertainment package around, or even a lot of the serious packages, that don't take advantage of that.

Just some indication of this is that the graphics memory in the IBM PC is right in the address space of the machine. What that means is you can directly manipulate those bits on the screen using any of the 8088 instructions. Particularly the string instructions can be used to great advantage to provide animation type effects up on the screen. We could not have done that on an 8-bit machine, because we would have used up that crucial 64K resource, whereas on the 8088 it is megabyte resource. We put it very high in memory, I think about three quarters of the way up, and so it is there anytime you want to use it.

Myself and someone else here wrote most of the demo programs used on the IBM machine in a matter of about three hours, because the extra versatility provided by directly manipulable graphics allowed us to put commands in BASIC that let you get at the full power of the machine very easily. In the case of the Apple, anybody who knows how to do really good high resolution graphics has to be a guru and so there is what I call a "bits and bytes barrier" to getting in and using the machine. And so to do a good program, you have to be both smart about bits and bytes, and creative enough to create the program. It is a rare individual who combines both of those talents.

In the IBM PC we have lowered the bits and bytes barrier so we will tap into some people with additional creativity and understanding of how to do whatever the particular need is. We are getting rid of the general need to get into the innards of the machine to make it really perform. The power of this machine is much more on the surface than an 8-bit machine could possibly deliver.

DAVID: Now that you are into the subject of graphics, tell us more.

BILL: Looking at the graphics, the things that I mean specifically are some of the simple verbs that have been added into the BASIC and I will highlight three of those. The CIRCLE statement is very straightforward, you simply state where the center of the circle is and what the radius is going to be and immediately the thing is drawn at an extremely rapid speed. Also, you've got a lot of other options, you can add it at the end of the statement, like start angle, and end angle, and aspect ratio. The default is simply to do a full round circle, and that is something that the user can get at and use, for example, to do pie charts.

Another statement is what we call PAINT. It is a very simple notion. You simply enter a point on the screen and its just like putting your paint brush down there and painting until you hit the edge of the screen or the border. Say you draw a white border and you want to paint until you hit white, so no matter what the figure you have there is, square or circle or crazy looking thing, it will use its paint brush and paint in until it finds those edges. As a default it paints in the same color as the edges, but if you provide an extra parameter you can paint with another color. So you could paint a white circle with a blue center, or, if you had some sort of a jagged line graph and you wanted to show it as an area, you find a point in the interior and it would paint that arbitrary figure.

PAINT is a single verb. It is quite simple and intuitive and yet its implementation is very hard. That brings through some of the power of this machine. You can paint a figure that's virtually the entire screen in about two seconds. Really, there is no way that that could have been done on an 8-bit machine. It may sound unimportant but when you really get into trying to do some of these new user interfaces, the so called Xerox Star-like interfaces that really are what is going to open up these machines to a wider user population, these graphics primitives are incredibly important. For example, when we put a little arrow up on the screen to point to things, we use a solid arrow, and to do that efficiently we have actually generated the thing with PAINT.

The final verb I wanted to mention is DRAW and this represents a philosophical

"If you sit down at the IBM machine and play with it a little while you will see that it performs significantly better than existing 8-bit machines."

decision we made a couple of years ago, which was that every time we put something new in BASIC there is a tendency to add a ton of verbs. In the case of graphics, where you are really adding verbs all of the time, and the user has a hard time remembering all of these verbs and each of them has its own individual syntax, and so that is a problem. The second problem is that if you use a bunch of verbs, then the description of a graphics object is not something you can read or write like a file. It is actually a program and so to move the embodiment of the graphics object around you have to move a program around. Well, that's a real pain, because in BASIC a program and data aren't treated uniformly and so you just get into big problems. What you'd really like is a simple way of using one of the data types already in BASIC to describe arbitrary graphics objects, and what we chose is the string data type. So now we have a simple single verb that gives you almost all the graphics capability and it is called DRAW.

Just to give one example, if you want to draw a box, you use the subverbs, which are R for right, L for left, U for up and D for down. So if I want a 10 × 10 square, I would enter DRAW, put a quote mark to indicate that it is a string, and go "R20 D20 L20 U20" and if I execute that it will draw the box. That is called Graphics Macro Language and the IBM PC is actually the third machine we have put that on. It has been extremely well received, and since using those strings you can write into a file or edit them or search for something inside them super easily.

That same concept has been used for music where it is called Music Macro Language, and so instead of DRAW you use PLAY. Enter PLAY "A, B, C" and it plays the notes A, B and C. It is true if somebody wants to specialize in them they have to learn the so-called macro language for that area, but it consists of really super simple commands and very self-contained.

Music is another case where I don't mean to pick on Apple—the only reason I use it is because it is an example of one of the most popular machines that has a lot of these capabilities and yet they are hard to get to. Once again, with music you have to be a real bits and bytes man to get that Apple to play any kind of decent tone. With the BASIC we have provided here, you can play something in legato, staccato or normal, just knowing a few simple characters that you type in under the control of BASIC. So we are pushing towards fulfilling the promise of these personal computers which is that anybody can just pick it up and use it—it's still not fulfilled but we are moving in the right direction.

"When we first get an extra resource, we don't know all the ways we are going to be able to take advantage of it."

DAVID: *We've been talking about things that IBM has done right which are significant. In your opinion, what are some of the things they have done wrong or not quite right?*

BILL: Well, you know in a way I am biased because of the depth of our involvement. I'd say it's a reasonably good balance, I mean in a way IBM is standing on the shoulder of experience that everybody else had in the industry—in a totally fair and good way, but it's not like 1976 when we didn't know what the market was and how to sell things. A lot of elements have been firmly established.

I have a wish list after we finish a project. I don't think cassette machines are super important and so I think the effort that was put into have a cassette interface wasn't worthwhile. I think everybody is going to run out of slots very quickly. The machine has a 5 slot limitation, but I suspect that an independent peripheral industry will start to do some combination cards that will reduce the pain of having a limited number of slots.

Everybody talks about how they'd like to have more disk space on the machine and of course I always like to see networking on a machine and nobody really has a good solution to that yet. It would be nice if there was a hard disk and I'm sure the independent vendors will come and put one of those on it.

It's possible to do a much better machine in a lot of ways from a hardware point of view. You could put a faster processor in. Intel's has the 8086. You could do a machine that is almost four times the performance. When Intel comes out with their 8087 chip, that will be a nice potential upgrade. I think IBM's Technical Reference Manual makes it clear they have an additional socket on there for that 8087 floating point processor but from my point of view, which is once again biased, the name of the game is software.

This machine will be significant because it will usher in a new generation of portable software which will be significantly better because of the speed, the address space, the instruction set, the underlying operating system, and the experience gained from the previous years.

I think five years from now the amount of software and the quality of the software on this machine will be incredible. It will dwarf what is available on mainframes, minicomputers and other machines.

DAVID: *I think we should talk a little bit about the operating system. Partly because I see a lot of confusion about MS-DOS and its relation to CP/M, and CP/M-86 more specifically. It seems that I read over and over again in the press that IBM has an operating system that is compatible with CP/M. Does it?*

BILL: Well, not really. There certainly is a lot of confusion about this issue. When IBM announced the machine on August 12 they said they'd be making available three operating environments. And the operating environment that we provided is know by IBM as Personal Computer DOS. We call it MS-DOS and Lifeboat Associates calls it SB-86. So we've got a lot of different names which adds a little bit to the confusion, but that's the operating system.

All of IBM's applications and languages that they're supporting run under it. In other words, VisiCalc only runs under PC DOS. The BASIC only runs under PC DOS, the Peachtree programs, and EasyWriter word processing package run under that. We've done some things there that are substantially different than has been done in CP/M. We did provide an upward migration path—in other words, we made it extremely easy if you've got source code and a translation package to move a CP/M-80 package up into the 8086 environment without worrying about the operating system interface. In other words, we emulate all the CP/M-80 calls because no doubt there is quite a wealth of CP/M-80 packages in existence. In fact the greatest installed base of CP/M-80 machines are the users of Microsoft softcards which plug into Apple computers. So we are probably as aware of that as anyone. Also I think we have more system software under CP/M-80 than any of the other vendors. So we made it possible to do that migration.

The move from 8-bit to 16-bit is an opportunity to improve things a great deal. CP/M-80 became a *de facto* standard in the 8-bit world. There is really no opportunity to change that—the 8-bit designers will essentially stop over the next year. The only chance to move up to a stronger

base is to grab this opportunity as we move into the new generation of processors.

Microsoft started out looking at 16-bit operating systems at the high end. About two years ago we went to Western Electric and licensed their Unix Operating System—which we have commercialized to a form known as Xenix. When IBM came along both from a technical point of view and other considerations it made sense for them to work with us on a new product we were doing which was a low-end operating system. So what we've got now is a family of operating systems with MSDOS at the low end and Xenix at the high end—really there's such a broad range of systems. From a single-user floppy system up to essentially a time sharing 16-bit system. We feel it is absolutely critical to have more than one operating system, although you have to have complete compatibility to move up along the line and add additional capability. That's what we have done with MSDOS.

DAVID: *Let's talk a little bit about IBM again. Who do you think the main customers are going to be for the IBM Personal Computer?*

BILL: I suspect that they will sell tons through their DPD sales force to large companies that have been looking at personalized work stations with local intelligence with a great deal of interest but too much fear to date.

The Apple II does not have enough communications capability and CRT capability to really be used in that mode. Until the IBM PC came along there was no product that could be offered to fill that need and I think that it is a huge market.

I've never heard any IBM estimates so I am just guessing here, but I think the majority of the sales will be through their DPD sales force. You know, Sears is doing a super job but they are only projecting five stores by the end of the year. No doubt Computerland will sell a lot of the machines but I doubt if they will be able to keep up with essentially the Fortune 500 demand from standard data processing departments.

DAVID: When do you think IBM will begin to sell through independent retailers?

BILL: All I know is what I read which is that towards the start of next year, they'll start to qualify additional retail vendors.

My understanding is that they will broaden their distribution. You know, IBM has to be admired for some of their conservatism. They only qualify the best and most professional groups to work together with them, because IBM is very afraid that somehow their overall corporate reputation is going to be hurt by what they are doing in this area.

DAVID: Still, IBM is doing some rather radical things, at least for IBM.

BILL: And it scares them that somehow that might hurt their image. So they went to Computerland, which is probably the leader in the independent dealer area. They gave *BYTE* magazine an initial exclusive on talking about the machine. They've really gone to the most established groups to do their work.

DAVID: How many machines do you think IBM will sell in 1982?

BILL: My guess is not based upon any inside information whatsoever but I think it will be not far from 200,000.

DAVID: Really?

BILL: If they can deliver them, the potential is there. I've heard numbers ranging anywhere from 100,000 to 150,000 so I am an optimist beyond the median point of that scale. They'll have to open up more distribution, though. I don't think Computerland can push through that many. And they may run into some production bottlenecks. There are a lot of outside vended parts on the machine and they are not going to compromise quality. Certainly at this point the machine is incredibly short, you know, we've got a ton on order and it is going to take a few months before they come in.

DAVID: Yes, we have the same problem. Let's move on to another topic, which we alluded to earlier. How does your soon-to-be-announced electronic spread sheet, Multiplan, relate to VisiCalc? Is it better?

BILL: Oh, certainly. It's a second generation spread sheet product. We'll be really going into that in our literature and it's a huge promotion thing for us—almost equal to all the promotion we have done for the entire company in its history, just for this one product. But, I'll just mention two things that are critical in Multiplan. The first is the use of naming. You are not put into a mode where you have to use "A10," "B9," "C14" and things like that, which you have to do with VisiCalc. If you want to say that taxes are 6% of sales then you say "taxes are .06 times sales." If you want the sum of all the profits you say "SUM (Profit)" and so we deal with data on a name basis which is the way people are used to dealing with it. The second thing is that we handle what we call Multisheet, which is a pretty obvious capability if you accept the analogy that these are spread sheet simulators. It is quite common to take numbers from, say, your cost sheet and your sales sheet and consolidate together. What you would really like is when you update the cost sheet it will carry over to the summary sheet. As soon as you look at the summary sheet, the information will be there. You don't have to type any commands or do any work every time you make the change to get the information over there. We have accommodated that capability.

One last thing, that I would like to mention also, is the way we have done the end-user interface. We've done away with slashes (/) and the need to know a lot of things about what is going on inside the package. For example, VisiCalc has a feature called "Order of Recalculation." The user has to think about does it go horizontally to recalculate or vertically to recalculate. Well, that's ridiculous. It's up to the computer to figure out the order of recalculation and not force you to figure out how you have to order your data so that things propagate through in the right order. That's a very technical thing.

DAVID: One thing you seem to be saying is that we are going to see a whole new set of application programs similar in concept to 8-bit programs only with a lot of improvements.

BILL: Right.

DAVID: Let's slip down the road five years. What are some of the real significant advances you see?

BILL: In five years the cost of computation will really be effectively decreased.

"We're still not at the stage where I'd tell my mother, or some naive person, just to go out and buy one of these machines."

We'll be able to put on somebody's desk, for an incredibly low cost, a processor with far more capability than you could ever take advantage of. Hardware in effect will become a lot less interesting. The total job will be in the software, and we'll be able to write big fat programs. We can let them run somewhat ineffeciently because there will be so much horsepower that just sits there. The real focus won't be who can cram it down in, or who can do it in the machine language. It will be who can define the right end-user interface and properly integrate the main packages. I expect over the next five years between us and others a heck of a job will get done. You'll be able to sit at your desk and do whatever it is you want to do with information or presenting data or interchanging data incredibly effectively. In other words, we will have changed the way people work.

At that time we'll just see the beginnings of the home information system, because it is so much harder to cost-justify that type of device. But I do feel that the "office of the future" will be the office of the present five years from now.

DAVID: What kind of mass storage device will machines have in five years?

BILL: Well, you'll probably still have local floppies in a lot of cases, but most of the storage size-wise will be in shared file servers—and although optical disk may have had an impact, even at present prices and capacities large (magnetic) disks would suffice. There are 300-megabyte disks down in the $10,000 to $15,000 range now. If you can spread it across 20 users—that is, with a good networking scheme—you could justify it. So, while there ought to be some improvement there, I don't think that we've got any bottleneck even today. Networking is probably one of the big challenges.

DAVID: How are you facing that challenge?

BILL: Well, we've designed a structure in MS-DOS that lets it work in a network environment in a very strong fashion—and it's substantially different than what Digital Research has defined for CP-Net. We're passing high level file calls down the network, through a tree-structured directory.

DAVID: What's the most satisfying experience you've had in this business to date?

BILL: I always sort of latch onto the most recent thing. This IBM project was a super-exciting, fun project. We were given, even for a small company, an incredible amount of latitude in changing how things got done as the project progressed. And we really were allowed to feel like some of the key work had been done here. And we had a really great interface with the people from the customer (IBM), even though they're as far away as they could be, down in Boca Raton. The night flight down there is not too much fun. We had a lot of fun together. We had an electronic mail linkup, and we'd send messages every day and we'd give each other a hard time about whichever group was behind on whatever they were responsible for. We loved to kid them about all the security—how we had to have locks, and sign things in, and use code names and stuff like that—but it was just part of the project camaraderie, really. When the thing finally got put together and we did the demo programs, everybody around here was enthused. That's something WE did!

I don't know how many people have read Tracy Kidder's new book *The Soul Of A New Machine,* but it was like that—and everybody really did get their just desserts of being recognized and knowing what part they put into it. People worked incredibly hard. I guess there was a kind of an anticlimax when I got a form letter from IBM a week after we'd finished the thing which said, "Dear Vendor. You've done a fine job." But they've apologized an appropriate number of times for that.

There'll be more projects. In fact, we're starting up one now which in its general concept should prove to be as exciting. And we're still not at the stage where I'd tell my mother, or some naive person, just to go out and buy one of these machines. In a couple of years we'll achieve that real peak—to fill that gap and feel like it's a real tool.

DAVID: It sounds like from what you're saying that you have probably had more influence on the final result of the machine than anyone, with possible exceptions at IBM.

BILL: Oh, that's absolutely the case. The people at IBM did a fantastic job and there's some super smart people there. I was very, very impressed with the team they put together. They used most of the people who had their own personal computers. Employees within IBM who have the oomph to go out and get their own personal computer and be kidded by their fellow workers, are in general a pretty good class of individuals. And a few of these people were just exceptional.

They were brought in from the company at large and they came down to Boca just for this project. We were the only vendor that understood what the project was about. Even up to the announcement most vendors were kept in the dark about the general scope and the general push of the thing. So we really enjoyed a really unique relationship. I don't think its flattering ourselves to say that I doubt that IBM has ever had such a relationship ever before. In fact, in their internal magazine—*Think*—they even mentioned the role that we played which was quite a thing for them to do. Other than this project, most outside vendors for IBM are really just providing their components and not super involved in how it fits in.

We developed a personal relationship with all those people that's equal to the closest project work we have done.

DAVID: Sounds like it was a lot of fun.

BILL: It was. Everybody around here enjoyed it a great deal. In a way, we always wanted there to be a definitive end to the thing, but even today there's some work going on. It's not like there is just one celebration. Boy, there has been some great...a lot of fun relaxation when we've hit various milestones. I don't know, the announcement was probably the best one because all the way through the project there was this aura that IBM couldn't even say to us that the project would be introduced. They always had to say, "You realize this may get cancelled any day and we'll just call you up and tell you to put all those confidential pieces of paper in a box and mail them back down here and don't call us again." I don't know how long that was really true, but that is really what they had to say to us. To know that the thing would really see the light of day and people would have a chance to evaluate what we had done really made us feel good.

We expect over the next year or two when people have really looked into the machine to see what it can do they will be increasingly impressed. Just like high resolution graphics on the Apple, there is a lot of capability there that will only unfold itself over a fair period of time. Some of that is the stuff we put in there and that will be neat.

I don't read about the TRS-80 any more because it does seem like a long time ago and in comparison it would be pretty easy to make fun of it, but the year or two after we did that project every time we would see somebody disassembling the BASIC or figuring out some little trick we thought it was really exciting.

It's the combination. Software is a great combination between artistry and engineering. When you finally get done and get to appreciate what you have done it is like a part of yourself that you've put together. I think a lot of the people here feel that way.

DAVID: That's quite a statement. Thank you for the interview.